Today we practice with ggplot2 graphics. Some (much?) of what you’ll be doing is not covered explicitly in the notes, so you will probably find the ggplot2 help at http://docs.ggplot2.org/current/ to be useful, as well as the graphics portion of the R cookbook site at http://www.cookbook-r.com/Graphs/. Of course Google is also quite helpful!

We will again use the gapminder data created by Jenny Bryan. To read these data into R and remind yourself of the structure use

```r
gapminder <- read.delim("http://www.stt.msu.edu/~melfi/STT301/data/gapminder.tsv", header = TRUE)
str(gapminder)
```

' 

 Note that prompts are not printed in the R code included in this document, which will make it easier for you to copy and paste the R code into your own R Markdown document or R session.

### Scatter Plots

Our main focus will be on scatter plots, although other types of graphical displays such as histograms and box plots will also make an appearance. In some cases the R code will be provided for you. In others the finished graphic will be provided, and your job will be to determine how to create that graphic. The next homework assignment will likely have that format, so this is a chance to practice.

First, we draw a simple scatter plot of life expectancy versus GDP. In this first code chunk we’ll need to load the ‘ggplot2’ library, since it’s not loaded by default in R.

```r
library(ggplot2)
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) + geom_point()
```
The form of the scatter plot suggests that a log transformation might be helpful. One possibility is to include the transformation in the initial `aes` specification:

```r
ggplot(data = gapminder, aes(x = log10(gdpPercap), y = lifeExp)) + geom_point()
```
That’s OK, but the scale on the x axis is now in "log" units. It might be better to use the original units, which we do next.

```r
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) + geom_point() + scale_x_log10()
```
An alert viewer of the graph will notice that the distance between 1000 and 10000 is the same as the distance between 10000 and 100000 and realize that the per capita GDP has been log scaled. But it doesn’t hurt to indicate this explicitly by changing the label on the x axis. As with many things in R (and in life) there are several ways to do this.

\[
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) + geom_point() + scale_x_log10(name = "per capita GDP (log10 scaled)")
\]
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) + geom_point() + scale_x_log10() + xlab("per capita GDP (log10 scaled)")
```r
ggplot(data = gapminder, aes(x = gdpPerCap, y = lifeExp)) + geom_point() + scale_x_log10() + labs(x = "per capita GDP (log10 scaled)"")
```
All are fine ways to change the x axis label. Let’s use the third method, also change the y axis label, and save this part of the graphic specification so we don’t have to keep retyping it.

```r
p <- ggplot(data = gapminder, aes(x = gdpPerCap, y = lifeExp)) +
  scale_x_log10() + labs(x = "per capita GDP (log10 scaled)",
y = "life expectancy")
p + geom_point()
```
**Exercise 1.** Using ‘p’ as a starting point, produce a scatter plot similar to the ones we have created, but with the countries from each of the five continents colored differently. Here is what the graphic should look like:

![Scatter plot with colored continents](image)

**Solution:**

```r
p + geom_point(aes(color = continent))
```
Exercise 2. (a) Add a least squares line to the scatter plot. Here is what the graphic should look like:
Solution:

```r
p + geom_point(aes(color = continent)) + stat_smooth(method = lm, se = FALSE)
```
(b) The least squares line is not very visible. Make it wider. Also change its color to green. Here is what the graphic should look like:
Solution:

```r
p + geom_point(aes(color = continent)) + stat_smooth(method = lm, se = FALSE, lwd = 2, color = "green")
```
(c) Instead of one least squares line summarizing all the countries, include separate least squares lines for each continent. Here is what the graphic should look like:
Solution:

```r
p + geom_point(aes(color = continent)) + stat_smooth(method = lm, se = FALSE, aes(color = continent), lwd = 1.5)
```
(d) Rather than least squares LINES, include a separate smoother for each continent. (Hint: You’ve been using `method = lm`. Change this to `method = loess`.) Here is what the graphic should look like:
Solution:

```r
p + geom_point(aes(color = continent)) + stat_smooth(method = loess, se = FALSE, aes(color = continent), lwd = 1.5)
```
Years, and specific countries

So far the focus has been on life expectancy versus per capita GDP. Next we investigate how life expectancy changes over time, and limit our attention to one or a few countries. First, here is the graphic for Rwanda.

```r
ggplot(data = subset(gapminder, country == "Rwanda"), aes(x = year, y = lifeExp)) + geom_line()
```
Probably adding the actual points being connected by the line segments would add clarity.

```r
ggplot(data = subset(gapminder, country == "Rwanda"), aes(x = year, y = lifeExp)) + geom_line() + geom_point()
```
Exercise 3. Choose five countries which you find interesting. Create a graphic similar to the one just created for Rwanda, but including the five countries of interest. Here is an example where I have chosen the five most populous countries (as of 2015).
Solution:

```r
five_countries <- c("China", "India", "United States", "Indonesia", "Brazil")
ggplot(data = subset(gapminder, country %in% five_countries),
       aes(x = year, y = lifeExp, color = country)) + geom_line() + geom_point()
```
Histograms, box plots, and facets

Consider the distribution of life expectancy. Here is a histogram of life expectancy.

```r
ggplot(data = gapminder, aes(x = lifeExp)) + geom_histogram()
```

*stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.*
By itself this doesn’t provide much information. One way to increase the usefulness is to plot separate histograms by continent. In `ggplot2`, facets provide a simple way to do this. There are two different types of facet specifications, `facet_grid` and `facet_wrap`. The faceting specification is `rows ~ columns`, and a dot `.` indicates that there should be no faceting on that dimension. A few examples will make this more clear.

```r
ggplot(data = subset(gapminder, year %in% c(1952, 1972) & continent != "Oceania"), aes(x = lifeExp)) + geom_histogram(aes(fill = continent)) + facet_wrap(continent ~ year)
```
ggplot(data = subset(gapminder, year %in% c(1952, 1972) & continent != "Oceania"), aes(x = lifeExp)) + geom_histogram(aes(fill = continent)) + facet_wrap(year ~ continent)
ggplot(data = subset(gapminder, year %in% c(1952, 1972) & continent != "Oceania"), aes(x = lifeExp)) + geom_histogram(aes(fill = continent)) + facet_grid(continent ~ year)
ggplot(data = subset(gapminder, year %in% c(1952, 1972) & continent != "Oceania"), aes(x = lifeExp)) + geom_histogram(aes(fill = continent)) + facet_grid(year ~ continent)
Exercise 4. Create a graphic with separate histograms of life expectancy for the four continents excluding Oceania, with the histograms arranged on top of one another. Use all years in the data set. The graphic should look like this:
Solution:

```r
ggplot(data = subset(gapminder, continent != "Oceania"), aes(x = lifeExp)) + geom_histogram(aes(fill = continent)) + facet_grid(continent ~ .)
```
Exercise 5. Side by side boxplots would help to understand the changes in life expectancy over time. Create the following boxplot. (Some hints: You’ll want to specify year as an x variable. But this isn’t enough. You also need to explicitly tell geom_boxplot that year is a grouping variable.)
Solution:

```
ggplot(gapminder, aes(x = year, y = lifeExp)) + geom_boxplot(aes(group = year))
```
END OF EXERCISE